

IN THE CLAIMS

1. (Currently Amended) A method for designing a computer program, comprising:

accessing a ~~plurality~~ substantially complete set of domain rules, each domain rule being ~~invariant~~; invariant and expressed as a narrative description;

defining a domain from the domain rules, the domain used to determine a problem space and a solution ~~space~~; space, substantially all solutions of the solution space can be generated from the substantially complete set of domain rules;

identifying one or more requirements of the domain from one or more supplemental sources;

generating a model that establishes the requirements of the domain;

accessing a plurality of business rules, each business rule being variable;

associating the one or more business rules with the model; and

generating a code corresponding to the model in order to design a computer program.

2. (Original) The method of Claim 1, further comprising:

collecting the domain rules and the business rules;

allocating the domain rules and the business rules to a plurality of use cases;

realizing the use cases; and

assessing the domain rules and the business rules in accordance with the realization.

3. (Original) The method of Claim 1, further comprising:

checking a syntax of the code; and

providing a notification if a syntax error is detected.

4. (Original) The method of Claim 1, further comprising:

checking a logical consistency of the code; and

providing a notification if a logical inconsistency is detected.

5. (Original) The method of Claim 1, further comprising:
checking a compatibility between the model and the code; and
providing a notification if an inconsistency is detected.

6. (Previously Presented) The method of Claim 1, wherein the model is
expressed according to a modeling language.

7. (Currently Amended) Logic for designing a computer program, the logic embodied in a computer-readable medium and when executed by a computer operable to:

access a ~~plurality~~ substantially complete set of domain rules, each domain rule being ~~invariant~~; invariant and expressed as a narrative description;

define a domain from the domain rules, the domain used to determine a problem space and a solution ~~space~~; space, substantially all solutions of the solution space can be generated from the substantially complete set of domain rules;

identify one or more requirements of the domain from one or more supplemental sources;

generate a model that establishes the requirements of the domain;

access a plurality of business rules, each business rule being variable;

associate the one or more business rules with the model; and

generate a code corresponding to the model in order to design a computer program.

8. (Original) The logic of Claim 7, further operable to:

collect the domain rules and the business rules;

allocate the domain rules and the business rules to a plurality of use cases;

realize the use cases; and

assess the domain rules and the business rules in accordance with the realization.

9. (Original) The logic of Claim 7, further operable to:

check a syntax of the code; and

provide a notification if a syntax error is detected.

10. (Original) The logic of Claim 7, further operable to:

check a logical consistency of the code; and

provide a notification if a logical inconsistency is detected.

11. (Original) The logic of Claim 7, further operable to:
check a compatibility between the model and the code; and
provide a notification if an inconsistency is detected.

12. (Previously Presented) The logic of Claim 7, wherein the model is expressed
according to a modeling language.

13. (Currently Amended) A system for designing a computer program, comprising:

a database operable to store a ~~plurality~~ substantially complete set of domain rules, each domain rule being ~~invariant~~; invariant and expressed as a narrative description; and

a server coupled to the database and operable to:

define a domain from the domain rules, the domain used to determine a problem space and a solution ~~space~~; space, substantially all solutions of the solution space can be generated from the substantially complete set of domain rules;

identify one or more requirements of the domain from one or more supplemental sources;

generate a model that establishes the requirements of the domain;

access a plurality of business rules, each business rule being variable;

associate the one or more business rules with the model; and

generate a code corresponding to the model in order to design a computer program.

14. (Original) The system of Claim 13, the server further operable to:

collect the domain rules and the business rules;

allocate the domain rules and the business rules to a plurality of use cases;

realize the use cases; and

assess the domain rules and the business rules in accordance with the realization.

15. (Original) The system of Claim 13, the server further operable to:

check a syntax of the code; and

provide a notification if a syntax error is detected.

16. (Original) The system of Claim 13, the server further operable to:

check a logical consistency of the code; and

provide a notification if a logical inconsistency is detected.

17. (Original) The system of Claim 13, the server further operable to:
check a compatibility between the model and the code; and
provide a notification if an inconsistency is detected.

18. (Previously Presented) The system of Claim 13, wherein the model is
expressed according to a modeling language.

19. (Currently Amended) A system for designing a computer program, comprising:

means for accessing a ~~plurality~~ substantiall y complete set of domain rules, each domain rule being ~~invariant~~; invariant and expressed as a narrative description;

means for defining a domain from the domain rules, the domain used to determine a problem space and a solution ~~space~~; space, substantially all solutions of the solution space can be generated from the substantially complete set of domain rules;

means for identifying one or more requirements of the domain from one or more supplemental sources;

means for generating a model that establishes the requirements of the domain;

means for accessing a plurality of business rules, each business rule being variable;

means for associating the one or more business rules with the model; and

means for generating a code corresponding to the model in order to design a computer program.

20. (Currently Amended) A method for designing a computer program, comprising:

collecting a ~~plurality~~ substantially complete set of domain rules for a military theory, allocating the domain rules to a plurality of use cases, realizing the use cases, assessing the domain rules in accordance with the realization, and accessing the domain rules, each domain rule being ~~invariant~~, invariant and expressed as a narrative description, the plurality of domain rules comprising a plurality of military theory domain rules setting an objective to destroy an enemy's combat forces;

defining a domain from the domain rules, the domain used to determine a problem space and a solution ~~space~~; space, substantially all solutions of the solution space can be generated from the substantially complete set of domain rules;

identifying one or more requirements of the domain from one or more supplemental sources;

generating a model that establishes the requirements of the domain;

displaying a plurality of business rules for the military theory, each business rule being variable, the plurality of business rules comprising a plurality of rules of engagement;

selecting one or more rules of engagement in response to a user selection;

customizing the one or more rules of engagement;

associating the one or more rules of engagement with the model;

associating the military theory domain rules with the model

generating a code corresponding to the model in order to design a computer program;

checking a syntax of the code, and providing a notification if a syntax error is detected;

checking a logical consistency of the code, and providing a notification if a logical inconsistency is detected; and

checking a compatibility between the model and the code, and providing a notification if an inconsistency is detected.

21. (Currently Amended) A method for managing rules for designing a computer program, comprising:

accessing a plurality of military theory rules for a military theory;

accessing a plurality of legislated laws associated with the military theory;

identifying military theory rules required by the laws as a plurality substantially complete set of domain rules of the military theory, each domain rule being invariant; invariant and expressed as a narrative description;

defining a domain from the domain rules, the domain used to determine a problem space and a solution ~~space~~; space, substantially all solutions of the solution space can be generated from the substantially complete set of domain rules;

identifying one or more requirements of the domain from one or more supplemental sources;

generating a model that establishes the requirements of the domain;

designating the other military theory rules as a plurality of business rules of the military theory, the business rules comprising a plurality of rules engagement, each business rule being variable; and

providing a rule of engagement from the rules of engagement in response to a request for the business rule.

22. (Previously Presented) The method of Claim 21, further comprising:

customizing the provided rule of engagement;

associating the customized rule of engagement with the model; and

generating a code corresponding to the model in order to design a computer program.

23. (Previously Presented) The method of Claim 21, further comprising:

associating the domain rules with the model; and

generating a code corresponding to the model in order to design a computer program.

24. (Original) The method of Claim 21, further comprising:
allocating the domain rules and the business rules to a plurality of use cases;
realizing the use cases; and
assessing the domain rules and the business rules in accordance with the realization.

25. (Currently Amended) A system for managing rules for designing a computer program, comprising:

a database operable to:

store a plurality of military theory rules for a military theory; and

store a plurality of legislated laws associated with the military theory; and

a server coupled to the database and operable to:

identify military theory rules required by the laws as a ~~plurality~~ substantially complete set of domain rules of the military theory, each domain rule being ~~invariant~~; invariant and expressed as a narrative description;

define a domain from the domain rules, the domain used to determine a problem space and a solution space; space, substantially all solutions of the solution space can be generated from the substantially complete set of domain rules;

identify one or more requirements of the domain from one or more supplemental sources;

generate a model that establishes the requirements of the domain;

designate the other military theory rules as a plurality of business rules of the military theory, the business rules comprising a plurality of rules engagement, each business rule being variable; and

provide a rule of engagement from the rules of engagement in response to a request for the business rule.

26. (Previously Presented) The system of Claim 25, wherein the server is further operable to:

customize the provided rule of engagement;

associate the customized rule of engagement with the model; and

generate a code corresponding to the model in order to design a computer program.

27. (Previously Presented) The system of Claim 25, wherein the server is further operable to:

associate the domain rules with the model; and
generate a code corresponding to the model in order to design a computer program.

28. (Original) The system of Claim 25, wherein the server is further operable to:
allocate the domain rules and the business rules to a plurality of use cases;
realize the use cases; and
assess the domain rules and the business rules in accordance with the realization.

29. (Currently Amended) Logic for managing rules for designing a computer program, the logic embodied in a computer-readable medium and when executed by a computer operable to:

access a plurality of military theory rules for a military theory;

access a plurality of legislated laws associated with the military theory;

identify military theory rules required by the laws as a ~~plurality~~ substantially complete set of domain rules of the military theory, each domain rule being ~~invariant~~; invariant and expressed as a narrative description;

define a domain from the domain rules, the domain used to determine a problem space and a solution ~~space~~; space, substantially all solutions of the solution space can be generated from the substantially complete set of domain rules;

identify one or more requirements of the domain from one or more supplemental sources;

generate a model that establishes the requirements of the domain;

designate the other military theory rules as a plurality of business rules of the military theory, the business rules comprising a plurality of rules engagement, each business rule being variable; and

provide a rule of engagement from the rules of engagement in response to a request for the business rule.

30. (Previously Presented) The logic of Claim 29, further operable to:

customize the provided rule of engagement;

associate the customized rule of engagement with the model; and

generate a code corresponding to the model in order to design a computer program.

31. (Previously Presented) The logic of Claim 29, further operable to:

associate the domain rules with the model; and

generate a code corresponding to the model in order to design a computer program.

32. (Original) The logic of Claim 29, further operable to:
allocate the domain rules and the business rules to a plurality of use cases;
realize the use cases; and
assess the domain rules and the business rules in accordance with the realization.

33. (Currently Amended) A system for managing rules for designing a computer program, comprising:

means for accessing a plurality of military theory rules for a military theory;

means for accessing a plurality of legislated laws associated with the military theory;

means for identifying military theory rules required by the laws as a plurality substantially complete set of domain rules of the military theory, each domain rule being ~~invariant;~~ invariant and expressed as a narrative description;

means for defining a domain from the domain rules, the domain used to determine a problem space and a solution ~~space;~~ space, substantially all solutions of the solution space can be generated from the substantially complete set of domain rules;

means for identifying one or more requirements of the domain from one or more supplemental sources;

means for generating a model that establishes the requirements of the domain;

means for designating the other military theory rules as a plurality of business rules of the military theory, the business rules comprising a plurality of rules engagement, each business rule being variable; and

means for providing a rule of engagement from the rules of engagement in response to a request for the business rule.

34. (Currently Amended) A method for managing rules for designing a computer program, comprising:

accessing a plurality of military theory rules for a military theory;

accessing a plurality of legislated laws associated with the military theory;

identifying military theory rules required by the laws as a plurality substantially complete set of domain rules of the military theory, each domain rule being invariant; invariant and expressed as a narrative description;

designating the other military theory rules as a plurality of business rules of the military theory, the business rules comprising a plurality of rules engagement, each business rule being variable;

defining a domain from the domain rules, the domain used to determine a problem space and a solution ~~space~~; space, substantially all solutions of the solution space can be generated from the substantially complete set of domain rules;

identifying one or more requirements of the domain from one or more supplemental sources;

generating a model that establishes the requirements of the domain;

allocating the domain rules and the business rules to a plurality of use cases;

realizing the use cases;

assessing the domain rules and the business rules in accordance with the realization;

storing the rules of engagement;

providing a rule of engagement from the stored rules of engagement in response to a request for the business rule;

customizing the provided rule of engagement;

associating the customized rule of engagement with the model;

associating the domain rules with the model; and

generating a code corresponding to the model in order to design a computer program.

35. (Canceled)

36. (Canceled)

- 37. (Canceled)
- 38. (Canceled)
- 39. (Canceled)
- 40. (Canceled)
- 41. (Canceled)
- 42. (Canceled)
- 43. (Canceled)
- 44. (Canceled)
- 45. (Canceled)
- 46. (Canceled)
- 47. (Canceled)
- 48. (Canceled)
- 49. (Canceled)
- 50. (Canceled)
- 51. (Canceled)
- 52. (Canceled)

53. (Canceled)

54. (Canceled)